Programmation fonctionnelle et logique

TP4

1. Calculer la longueur d'une liste.

```
size([],0).
size([T|R],S) :- size(R,U), S is U+1.
```

2. Calculer la somme des éléments d'une liste.

```
sum([],0).
sum([T|R],M) :- sum(R,S), M is T+S.
```

3. Calculer la moyenne des éléments d'une liste.

```
moyenne(L,M) :- sum(L,S), size(L,A), M is S/A.
```

4. On appelle carré parfait un carré à cinq chiffres, dont la somme des chiffres est égale à 27 et dont les chiffres des milliers, des dizaines, des unités et des centaines sont consécutifs (et croissants dans l'ordre donné).

```
chiffre(X) :- between(0,9,X).

carre_parfait(X) :-
   chiffre(A), chiffre(B),
   D is B+1, E is D+1, C is E+1,
   X is A*10000 + B*1000 + C*100 + D*10  + E,
   A + B + C + D + E =:= 27,
   between(sqrt(X)-1,sqrt(X),Y),
   X =:= Y*Y.
```

5. Un carré magique est un tableau de trois cases de côtés dont la somme des lignes et des colonnes est constante. Écrire un programme qui vérifie si un carré est magique de somme une valeur donnée.

```
tous_differents([]).
tous_differents([T|L]) :- paselt(T,L), tous_differents(L).
paselt(X,[]).
paselt(X,[Y|L]) :- paselt(X,L), X =\= Y.

estelt(X,[X|Y]).
estelt(X,[Y|L]) :- estelt(X,L).
chiffre(T) :- estelt(T,[0,1,2,3,4,5,6,7,8,9]).

nombre(T) :- chiffre(X), chiffre(Y), chiffre(Z), T is X*100 + 10*Y + Z.

tous_nombres([],S).
tous_nombres([T|L],S) :- between(1,S,T), tous_nombres(L,S).

carre_magique(A,B,C,D,E,F,G,H,I,S) :-
tous_nombres([A,B,C,D,E,F,G,H,I],S),
S =:= A + B + C,
```

```
S =:= D + E + F,
S =:= G + H + I,
S =:= A + D + G,
S =:= B + E + H,
S =:= C + F + I,
tous_differents([A,B,C,D,E,F,G,H,I]).
```

6. Un carré magique est parfait si la somme de ses 6 diagonales est constante et égale à la somme de ses lignes.

```
carre_magique_parfait(A,B,C,D,E,F,G,H,I,S) :-
tous_nombres([A,B,C,D,E,F,G,H,I],S),
S =:= A + B + C,
S =:= D + E + F,
S =:= G + H + I,
S =:= A + D + G,
S =:= B + E + H,
S =:= C + F + I,
S =:= A + E + I,
S =:= B + F + G,
S =:= C + D + H,
S =:= G + E + C,
S =:= H + F + A,
S =:= D + B + I,
tous_differents([A,B,C,D,E,F,G,H,I]).
```